FxPlug Overview

Introduction for plug-in developers

Session 305

These are confidential sessions—please refrain from filming or taking pictures.

Apple Applications

FxPlug Overview

Introduction for plug-in developers

Dave Howell
FxPlug Engineering Manager

http://developer.apple.com/wwdc2005/

What You'll Learn

- What is the FxPlug SDK?
- Why did we develop FxPlug?
- How can you write FxPlug plug-ins?

Introduction to FxPlug

What is FxPlug?

- FxPlug: new visual-effects plug-in architecture
- Created by Apple Pro apps team
- First host app: Motion 2.0

Introduction to FxPlug

Why a new plug-in architecture?

- Pro apps features
- Hardware acceleration
- No more cross-platform lowest common denominator
- Leverage Apple strengths
  - OpenGL
  - Core Image
  - Interface Builder
  - Objective-C

Introduction to FxPlug

Why Objective-C?

- C++ elegance without the fragility
- Stable ABI
- Access to AppKit frameworks
- Interface Builder integration

Introduction to FxPlug

Why write FxPlug plug-ins?

- Over 350,000 potential seats (Motion and Final Cut users)
- Extreme performance
- Custom UI, based on Cocoa and Interface Builder
- OpenGL-based on-screen controls

Demo

dvMatte

Ben Syverson
dvGarage

http://developer.apple.com/wwdc2005/

How to Write an FxPlug Plug-in

Dave Howell

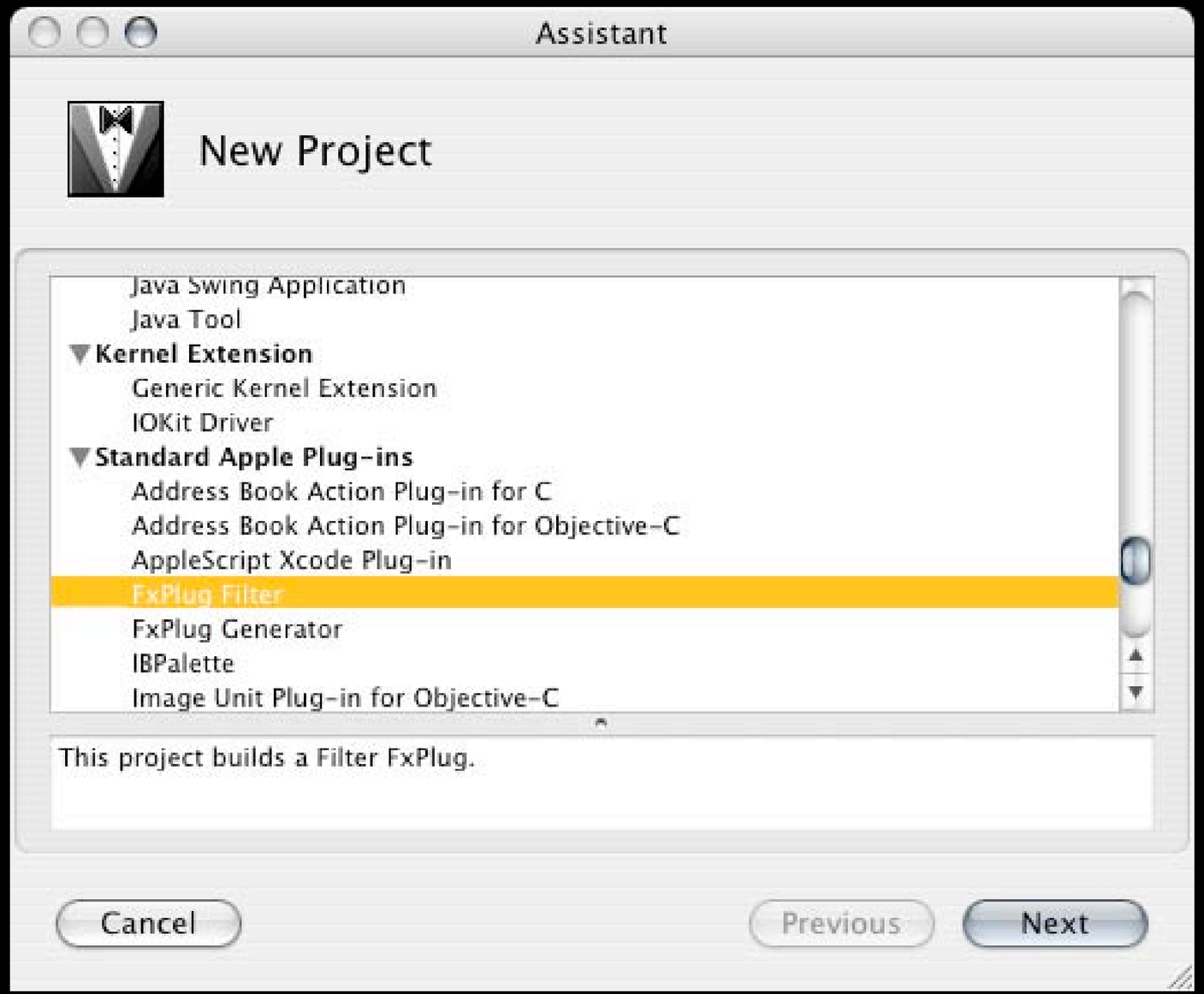
http://developer.apple.com/wwdc2005/

How to Write an FxPlug Plug-in

Overview

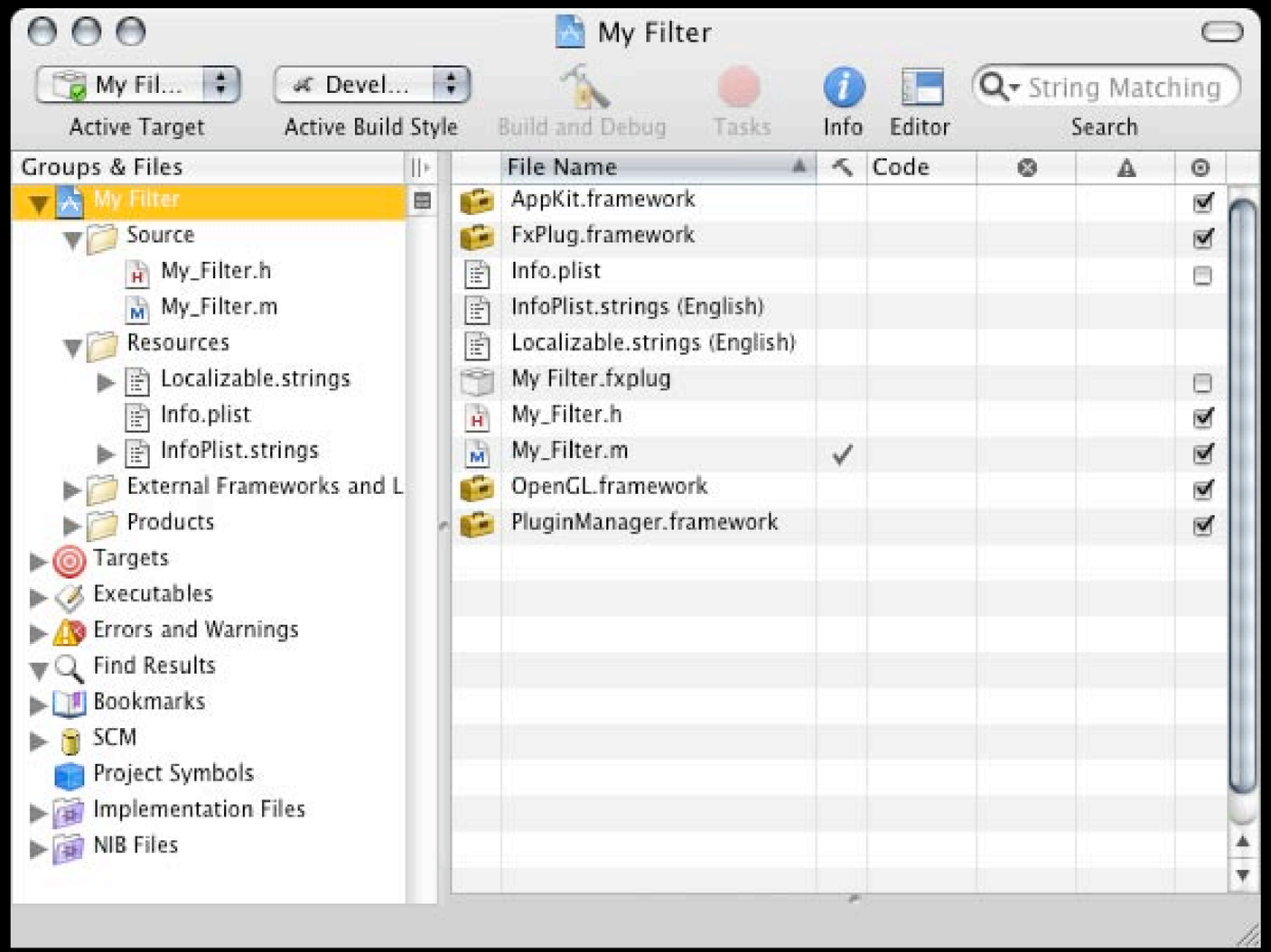
- Choose Xcode template
  - FxPlug filter or FxPlug generator
- Create unique IDs
- Customize source code
  - Edit parameter list
  - Edit render method
    - CPU and/or GPU
- Build and Install
- Test in Motion

Choose Xcode Template



Choose Xcode Template

Choose Xcode Template



Choose Unique IDs

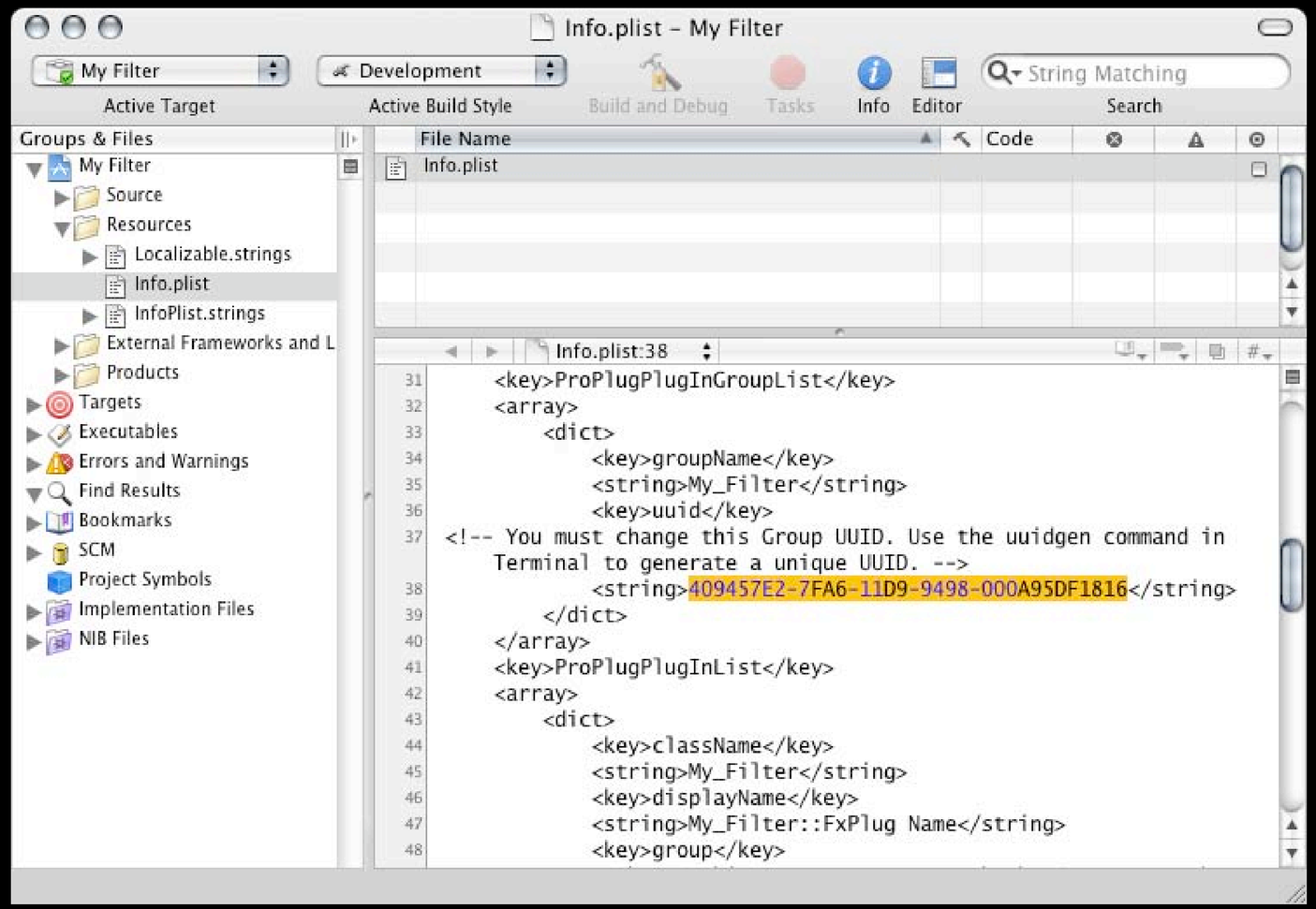
Replace UUIDs in Info.plist

- TIP: add InsertUUID Xcode Script
  - Described in documentation
  - /Library/Application Support/Apple/Developer Tools/Scripts/10-User Scripts/05-InsertUUID.sh

```
#! /bin/sh
#
# InsertUUID.sh - Invoke uuidget tool to generate and insert a UUID.
#
# %%%{PBXName=Insert UUID}%%%
# %%%{PBXInput=None}%%%
# %%%{PBXOutput=ReplaceSelection}%%%
# %%%{PBXKeyEquivalent=@$U}%%%
#
echo -n `uuidgen`
```

Choose Unique IDs

Replace UUIDs in Info.plist



Choose Unique IDs

Replace UUIDs in Info.plist

- Group list

```
<key>ProPlugPlugInGroupList</key>
<array>
  <dict>
     <key>groupName</key>
     <string>My_Filter</string>
     <key>uuid</key>
     <string>409457E2-7FA6-11D9-9498-000A95DF1816</string>
  </dict>
</array>
```

Choose Unique IDs

Replace UUIDs in Info.plist

- Plug-in list

```
<key>ProPlugPlugInList</key>
<array>
  <dict>
     <key>className</key>
     <string>My_Filter</string>
     <key>displayName</key>
     <string>My_Filter::FxPlug Name</string>
     <key>group</key>
     <string>409457E2-7FA6-11D9-9498-000A95DF1816</string>
     <key>uuid</key>
     <string>E5F40FB4-C66C-498E-8D61-0B2E23A40364</string>
        ⋮
  </dict>
</array>
```

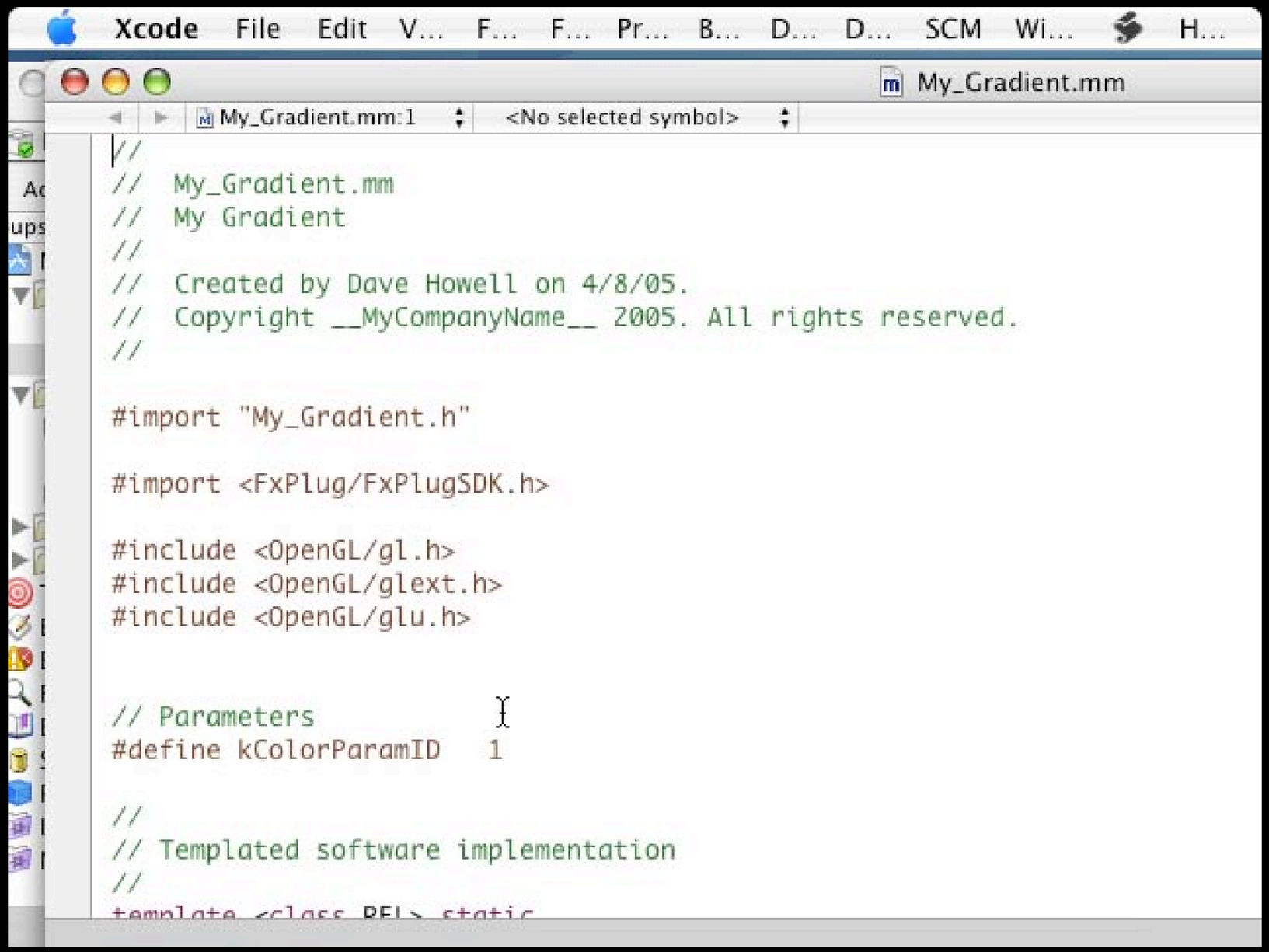
Choose Unique IDs

Replace UUIDs in Info.plist

- Plug-in list

```
<key>ProPlugPlugInList</key>
<array>
  <dict>
     <key>className</key>
     <string>My_Filter</string>
     <key>displayName</key>
     <string>My_Filter::FxPlug Name</string>
     <key>group</key>
     <string>409457E2-7FA6-11D9-9498-000A95DF1816</string>
     <key>uuid</key>
     <string>E5F40FB4-C66C-498E-8D61-0B2E23A40364</string>
        ⋮
  </dict>
</array>
```

Customize Source Code



Build and Install

Build and Install

Installation

- Install plug-ins in /Library/Plug-Ins/FxPlug or ~/Library/Plug-Ins/FxPlug
- DEBUGGING TIP: create symbolic link to point to built product

```
% ln -s -f ~/build/MyFilter.fxplug /Library/Plug-ins/FxPlug/MyFilter.fxplug
```

Test in Motion

FxPlug Plug-ins

ProPlug

- Pro Apps extension of NSBundle
- Defined by PluginManager framework
- Static or dynamic plug-in registration
- Plug-in classes conform to plug-in protocols
- Host app provides objects that conforms to host API protocols

ProPlug

PluginManager framework

- Protocols
  - PROPlugInRegistering (for dynamic registration)

```
#import <PluginManager/PROPlugInBundleRegistration.h>
```

ProPlug

PluginManager framework

- Protocols
  - PROPluginRegistering (for dynamic registration)

```
#import <PluginManager/PROPlugInBundleRegistration.h>
```

  - PROAPIAccessing (for retrieving host APIs)

```
#import <PluginManager/PROAPIAccessing.h>
```

ProPlug

Registration

- Static
  - Info.plist defines group(s) and plug-in(s)

```
<key>ProPlugDynamicRegistration</key>
  <false/>
<key>ProPlugPlugInGroupList</key>
  <array> <dict> ... </dict> </array>
<key>ProPlugPlugInList</key>
  <array> <dict> ... </dict> </array>
<key>ProPlugProtocolList</key>
  <array> <dict> ... </dict> </array>
```

ProPlug

Registration

- Static
  - Info.plist defines group(s) and plug-in(s)

- Dynamic
  - Plug-in's principal class conforms to PROPlugInRegistering protocol

```
+ (id)sharedInstance;
- (BOOL)shouldLoadFirstInstanceOfPlugInWithError:(NSError **)error;
- (NSArray *)registeredPlugInGroupsWithError:(NSError **)error;
- (NSArray *)requestedProtocolsWithError:(NSError **)error;
- (NSArray *)registeredPlugInsWithError:(NSError **)error;
```

ProPlug

Host APIs

- Get API Manager at initialization

```
- (id)initWithAPIManager:(id<PROAPIAccessing>)apiManager
{
  _apiManager = apiManager;
  return self;
}
```

ProPlug

Host APIs

- Get API Manager at initialization

```
- (id)initWithAPIManager:(id<PROAPIAccessing>)apiManager
{
   _apiManager = apiManager;
   return self;
}
```

- Use API Manager to request host API access

```
   id api = [_apiManager apiForProtocol:
                @protocol(FxParameterRetrievalAPI)];
   if ( api != NULL )
      [api getFloatValue:&val fromParm:kParmID atTime:renderInfo.frame];
```

FxPlug SDK

- Plug-ins
  - FxFilter, FxGenerator
- Optional custom UI support
  - FxOnScreenControl, FxCustomParameterUI
- Host APIs
  - FxParameterAPI, FxOptionalParameterAPI
  - FxOnScreenControlAPI
  - FxTemporalImageAPI
  - FxLayerInfoAPI
- Images
  - FxImage, FxBitmap, FxTexture

Parameters

Parameter list creation

- Implement -addParameters method
- Use FxParameterCreationAPI methods to add each parameter

```
- (BOOL)addParameters
{
   id parmAPI = [_apiManager apiForProtocol:
              @protocol(FxParameterCreationAPI)];
   if ( parmAPI != NULL )
   {
      [parmAPI addFloatSliderWithName:...];
      [parmAPI addPointParameterWithName:...];
      return YES;
   }
   else return NO;
}
```

Parameters

Parameter types

- Float slider

```
- (BOOL)addFloatSliderWithName:(NSString *)name
                       parmId:(UInt32)parmId
                 defaultValue:(double)defaultValue
                 parameterMin:(double)min
                 parameterMax:(double)max
                    sliderMin:(double)sliderMin
                    sliderMax:(double)sliderMax
                        delta:(double)sliderDelta
                    parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- Integer slider

```
- (BOOL)addIntSliderWithName:(NSString *)name
                      parmId:(UInt32)parmId
                defaultValue:(int)defaultValue
                parameterMin:(int)min
                parameterMax:(int)max
                   sliderMin:(int)sliderMin
                   sliderMax:(int)sliderMax
                       delta:(int)sliderDelta
                   parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

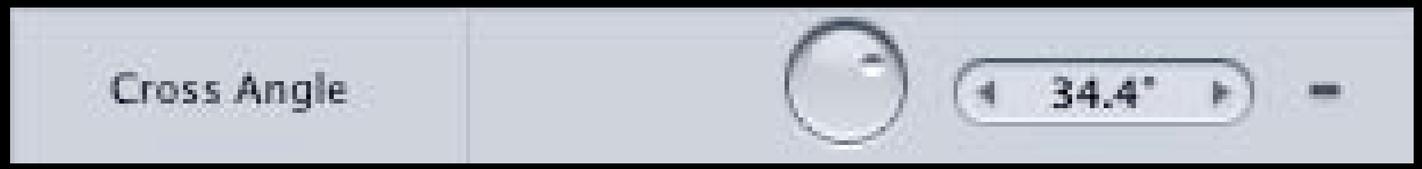
- Toggle button

```
- (BOOL)addToggleButtonWithName:(NSString *)name
                         parmId:(UInt32)parmId
                   defaultValue:(BOOL)defaultValue
                     parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- Angle slider

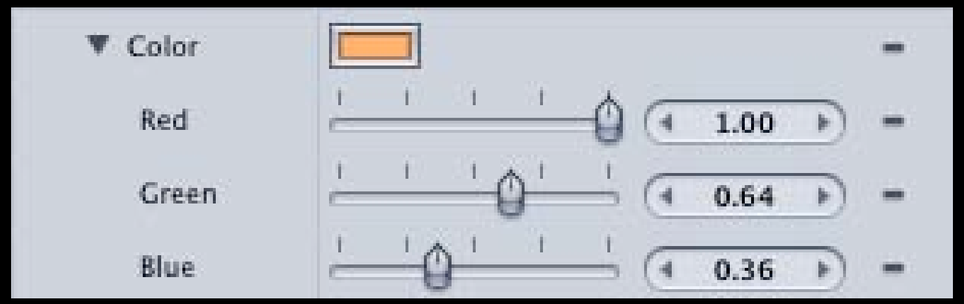



```
- (BOOL)addAngleSliderWithName:(NSString *)name
                       parmId:(UInt32)parmId
                 defaultValue:(double)defaultValue
                 parameterMin:(int)min
                 parameterMax:(int)max
                    parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- RGB color




```
- (BOOL)addColorParameterWithName:(NSString *)name
                           parmId:(UInt32)parmId
                       defaultRed:(double)red
                     defaultGreen:(double)green
                      defaultBlue:(double)blue
                        parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

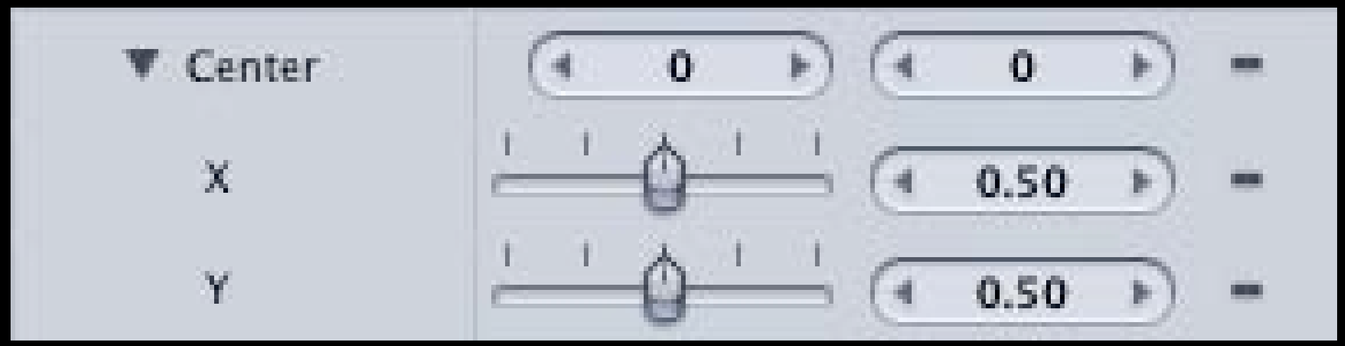
- ARGB color

```
- (BOOL)addColorParameterWithName:(NSString *)name
                           parmId:(UInt32)parmId
                       defaultRed:(double)red
                     defaultGreen:(double)green
                      defaultBlue:(double)blue
                     defaultAlpha:(double)alpha
                        parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- 2D point




```
- (BOOL)addPointParameterWithName:(NSString *)name
                           parmId:(UInt32)parmId
                         defaultX:(double)defaultX
                         defaultY:(double)defaultY
                        parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- Popup menu

```
- (BOOL)addPopupMenuWithName:(NSString *)name
                      parmId:(UInt32)parmId
                defaultValue:(UInt32)defaultValue
                 menuEntries:(NSArray *)entries
                   parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

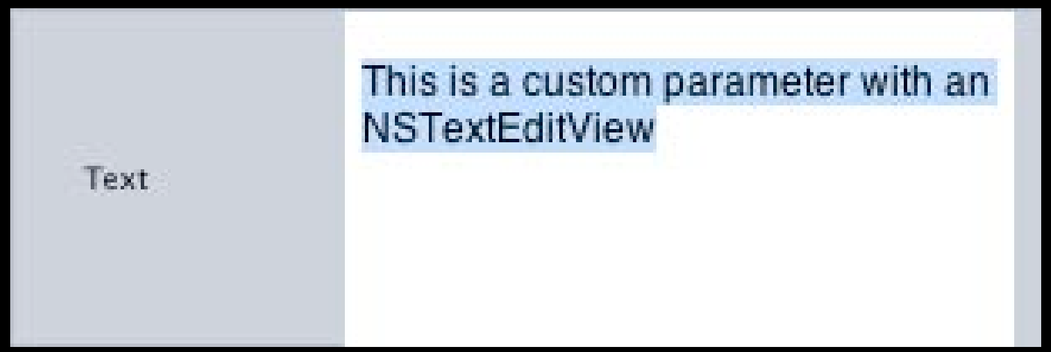
- Image

```
- (BOOL)addImageReferenceWithName:(NSString *)name
                           parmId:(UInt32)parmId
                        parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- Custom

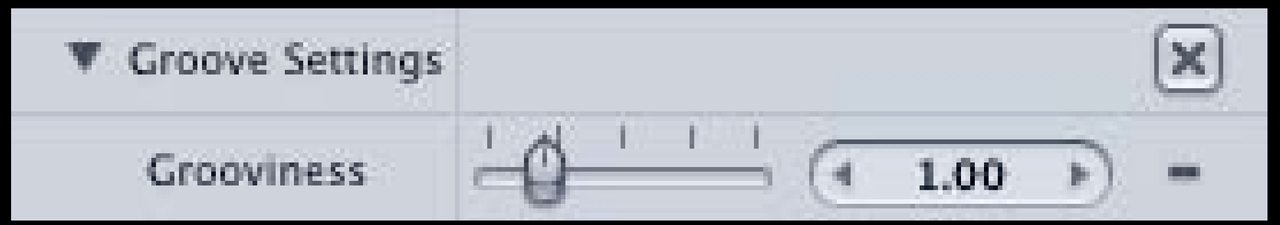



```
- (BOOL)addCustomParameterWithName:(NSString *)name
                            parmId:(UInt32)parmId
                      defaultValue:(id<NSCoding>)defaultValue
                         parmFlags:(FxParameterFlags)flags;
```

Parameters

Parameter types

- Group




```
- (BOOL)startParameterSubGroup:(NSString *)name
                       parmId:(UInt32)parmId
                   parmFlags:(FxParameterFlags)flags;


- (BOOL)endParameterSubGroup;
```

Parameters

Parameter types

- Gradients and histograms

Related session

FxPlug in Depth

Session 308
Friday 10:30AM

Parameters

Getting values

```
id api = [_apiManager apiForProtocol:@protocol(FxParameterRetrievalAPI)];

[api getFloatValue:&fVal fromParm:id1 atTime:f];
[api getIntValue:&iVal fromParm:id4 atTime:f];
[api getBoolValue:&bVal fromParm:id3 atTime:f];
[api getRedValue:&r GreenValue:&g BlueValue:&b fromParm:id4 atTime:f];

// etc.
```

Parameters

Setting values

```
id api = [_apiManager apiForProtocol:@protocol(FxParameterSettingAPI)];

[api setFloatValue:fVal toParm:id1 atTime:f];
[api setIntValue:iVal toParm: id4 atTime:f];
[api setBoolValue:bVal toParm: id3 atTime:f];
[api setRedValue:r GreenValue:g BlueValue:b toParm: id4 atTime:f];

// etc.
```

Rendering

Pete Warden
Graphics Coder

http://developer.apple.com/wwdc2005/

Rendering

Writing a GPU filter

- Write an ARB_fragment_program

```
static const char *_myARBFragmentProgram =
     “!!ARBfp1.0\n”
     “TEMP inSample;\n”
     “TEX inSample, fragment.texcoord[0], texture[0], RECT;\n”
     “MUL result.color, inSample, program.local[0];\n”
     “END”;
```

Rendering

Writing a GPU filter

- Write an ARB_fragment_program
- Write a pixel shader using the ARB_fragment_program

```
static GLUint _programID;

glGenProgramsARB( 1, &_programID );
glBindProgramARB( GL_FRAGMENT_PROGRAM_ARB, _programID );
glProgramStringARB( GL_FRAGMENT_PROGRAM_ARB, GL_PROGRAM_FORMAT_ASCII_ARB,
     strlen( _myARBFragmentProgram ), _myARBFragmentProgram );

glBindProgramARB( GL_FRAGMENT_PROGRAM_ARB, _programID );
```

Rendering

Writing a GPU filter

- Write an ARB_fragment_program
- Write a pixel shader using the ARB_fragment_program
- Draw OpenGL quad with shader
  - Bind input image as a texture
  - Given coordinates will cover output image
- Host app sets everything up for you

Related session

FxPlug in Depth

Session 308
Friday 10:30AM

Rendering

FxImage

- FxImage: base class
- FxTexture
  - OpenGL PBuffer
- FxBitmap
  - RAM-based bitmap image

Rendering

FxBitmap

- Pixel formats
  - 8-bit integer ARGB
  - 16-bit integer ARGB
  - 32-bit float ARGB
- Premultiplied alpha

Rendering

FxTexture

- Coordinates
  - Always ask for them
- Thin wrapper for OpenGL textures
  - Do bind/enable to use
  - Can get ID
- Premultiplied alpha

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

FxPlug in Depth	Session 308 Friday 10:30AM
FxPlug Development	Pro Audio and Video Connection Friday 12:45PM

Whom to Contact

Patrick Collins
Technology Manager
patrick@apple.com

Pro App SDK Group

proappsdk@group.apple.com

Q&A

Patrick Collins
Technology Manager
patrick@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference